

Task 2

Table of Contents

1 Bisection $f(x) = \sin(x)\sinh(x)$; Limits = [1 4] [3].....	1
1 Bisection $f(x) = (R^2-x^2)^{1/2} - x\tan(x)$ [3].....	3
2 Newtons $f(x) = e^x - (1+x+(x^2/2))$ [3].....	5
2 Newtons $f(x) = 10\log(x)-x$ [3].....	7
3 Newtons Intersect [6].....	9
4 Bracketing Bisect [6].....	13
4 Bracketing Newton [6].....	15

1 Bisection $f(x) = \sin(x)\sinh(x)$; Limits = [1 4] [3]

```
clear;
close all;
clc;

a = 1;
b = 4;
x = linspace(a,b,100);

function [function_value] = my_function(x)
    function_value = sin(x).*sinh(x); %The answer is (pi,0)
end

f_of_x=my_function(x);
x_guess_l = 1;
x_guess_u = 4;
tol= 1e-6;
imax = 50;
counter = 1;
hold on
while (abs(x_guess_u-x_guess_l)>tol && counter <imax)
    f_at_guess_l = my_function(x_guess_l);
    f_at_guess_u = my_function(x_guess_u);

    %Calculate mid-point
    mid_point = (x_guess_l + x_guess_u)/2;
    f_at_mid_point = my_function(mid_point);

    % % Output to screen
    % fprintf('ITERATION %i \r',counter);
    % fprintf('Lower guess = %10.6f \r',x_guess_l);
    % fprintf('Upper guess = %10.6f \r',x_guess_u);
    % fprintf('Mid-point = %10.6f \r \r',mid_point);

    plot([x_guess_l x_guess_l],[0 f_at_guess_l],'r-')
    plot([x_guess_u x_guess_u],[0 f_at_guess_u],'r-')
```

```

% fprintf('Value of function = %6.4f \r',f_at_guess_l)
% fprintf('Value of function = %6.4f \r',f_at_guess_u)
if f_at_mid_point * f_at_guess_u <= 0
    x_guess_l = mid_point;
else
    x_guess_u = mid_point;
end
counter = counter + 1;
end
figure(1);
plot(x,f_of_x);
% Plot midpoint
plot([1 4],[0 0],'r')    % plot y=0 axis
figure(1)
plot(mid_point,0,"o")
fprintf('Mid Point = %6.4f \r',mid_point);

```

Mid Point = 3.1416

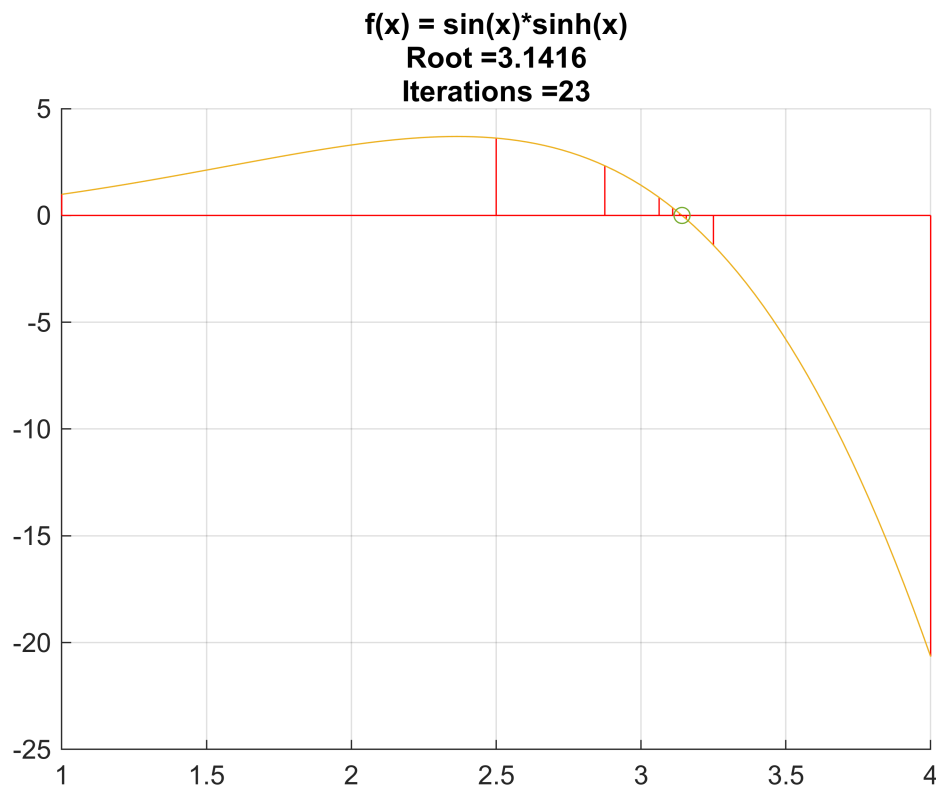
```

title(num2str(mid_point))

title({'f(x) = sin(x)*sinh(x)', ['Root =',num2str(mid_point)],['Iterations
=',num2str(counter)]})
grid on

hold off

```



1 Bisection $f(x) = (R^2 - x^2)^{1/2} - x \tan(x)$ [3]

```
clear;
close all;
clc;

R = 10;
a = 4;
b = 4.7;
x = linspace(a,b,100);

figure(2);
hold on

function [function_value] = my_second_function(x,R)
    R=10;
    function_value = sqrt((R.^2)-(x.*x))-x.*tan(x);%The answer is (4.2711,0)
end

f_of_x=my_second_function(x,R);
x_guess_l = 4;
x_guess_u = 4.7;
tol= 1e-6;
imax = 50;
counter = 1;
```

```

while (abs(x_guess_u-x_guess_l)>tol && counter <imax)
    f_at_guess_l = my_second_function(x_guess_l);
    f_at_guess_u = my_second_function(x_guess_u);

    %Calculate mid-point
    mid_point = (x_guess_l + x_guess_u)/2;
    f_at_mid_point = my_second_function(mid_point);

    plot([x_guess_l x_guess_l],[0 f_at_guess_l],'r-')
    plot([x_guess_u x_guess_u],[0 f_at_guess_u],'r-')

    if f_at_mid_point * f_at_guess_u <= 0
        x_guess_l = mid_point;
    else
        x_guess_u = mid_point;
    end
    counter = counter + 1;
end

plot(x,my_second_function(x));
plot([4 4.7],[0 0],'r') % plot y=0 axis
plot(mid_point,0,"o")
fprintf('Mid Point = %6.4f \r',mid_point)

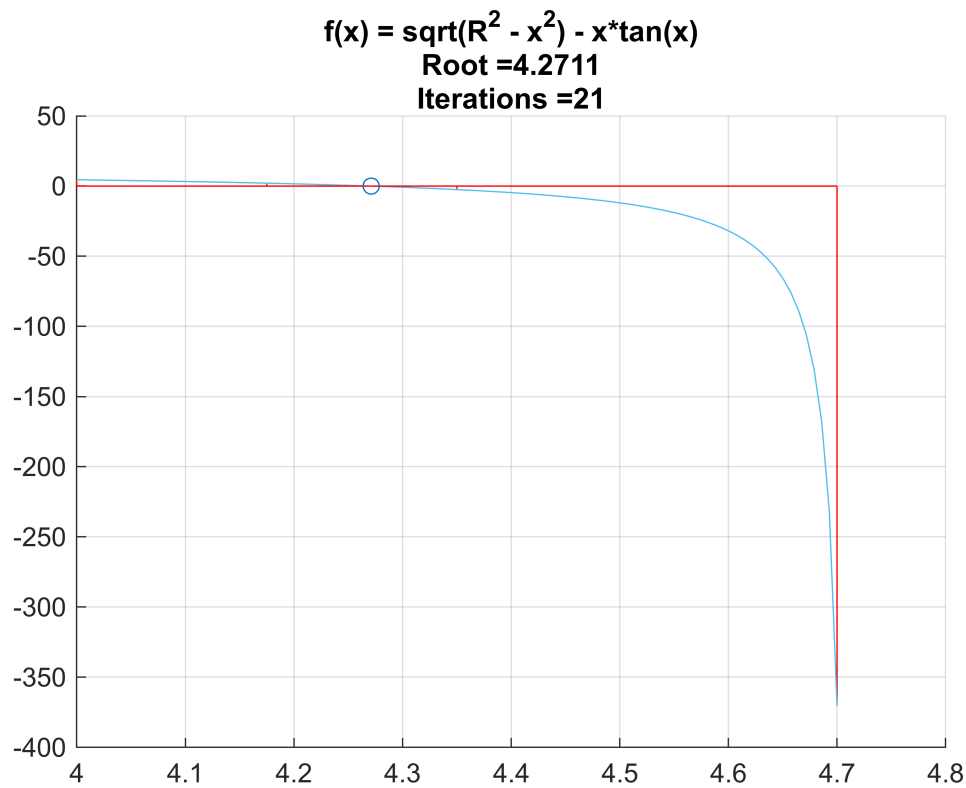
```

Mid Point = 4.2711

```

title({'f(x) = sqrt(R^2 - x^2) - x*tan(x)', ['Root =',num2str(mid_point)],
['Iterations =',num2str(counter)]})
grid on
hold off

```



2 Newtons $f(x) = e^x - (1 + x + (x^2/2))$ [3]

```
close;
clear;
clc;

e = 2.718281828459045235360;

x = linspace(0,5,100);
tolerance= 0.000001;
max_count = 20;
counter = 1;

function [function_value] = newton(x,e)
    e = 2.718281828459045235360;
    function_value = (e.^x) - (1 + x + ((x.^2)/2));
end

h_of_x=newton(x,e);
h_of_x_prime = (e.^x)-x-1;

x_guess_first = 5; %Initial guess
x_guess_second = x_guess_first - (h_of_x/h_of_x_prime); %Newtons Method for second
guess
```

hold on

```
while (abs(x_guess_first-x_guess_second) > tolerance && counter < max_count)
```

```
    f_at_guess_first = newton(x_guess_first);  
    f_at_guess_second = newton(x_guess_second);
```

```
    x_next = x_guess_second-f_at_guess_second * ( (x_guess_second -  
x_guess_first) / (f_at_guess_second - f_at_guess_first));
```

```
    % fprintf("ITERATION %i \r", counter);  
    % fprintf("First Guess = %10.6f \r", x_guess_first);  
    % fprintf("Second Guess = %10.6f \r", x_guess_second);  
    % fprintf("Next guess = %10.6f \r", x_next);
```

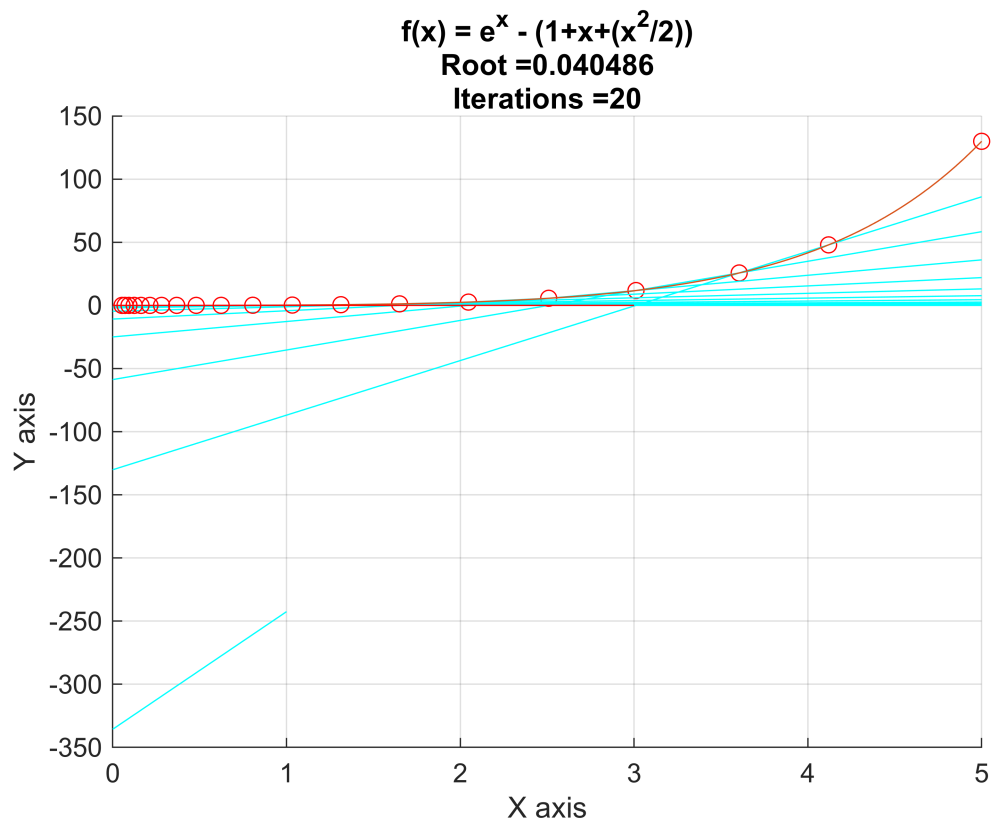
```
    x_limits=get(gca,"XLim");  
    m = (f_at_guess_second-f_at_guess_first)/(x_guess_second-x_guess_first);  
    n = -x_guess_second*m + f_at_guess_second;  
    y1 = m*x_limits(1) + n;  
    y2 = m*x_limits(2) + n;  
    plot(x_guess_first,f_at_guess_first,'ro')  
    plot(x_guess_second,f_at_guess_second,'ro')  
    plot([x_limits(1) x_limits(2)],[y1 y2],'c')
```

```
    x_guess_first=x_guess_second;  
    x_guess_second = x_next;
```

```
    counter = counter+1;
```

end

```
x_next;  
plot(x,h_of_x)  
%plot(x,h_of_x_prime)  
title(num2str(x_next))  
xlabel("X axis")  
ylabel("Y axis")  
plot([0 3],[0 0],'r') % plot y=0 axis  
title({'f(x) = e^x - (1+x+(x^2/2))', ['Root = ',num2str(x_next)], ['Iterations  
= ',num2str(counter)]})  
grid on  
hold off
```



2 Newtons $f(x) = 10\log(x)-x$ [3]

```
close;
clear;
clc;

x = linspace(0.001,2,100);

tolerance = 0.000001;
max_count = 20;
counter = 1;

function [function_value] = newto(x)
    function_value = (10 * log10(x)) - x;
end

h_of_x = newto(x);
h_of_x_prime = (10 ./ x) - 1;

x_guess_first = 1;
x_guess_second = x_guess_first - (h_of_x / h_of_x_prime);

hold on
figure(1)
```

```

while (abs(x_guess_first-x_guess_second) > tolerance && counter < max_count)

    f_at_guess_first = newto(x_guess_first);
    f_at_guess_second = newto(x_guess_second);

    x_next = x_guess_second-f_at_guess_second * ( (x_guess_second -
x_guess_first) / (f_at_guess_second - f_at_guess_first));

    % fprintf("ITERATION %i \r", counter);
    % fprintf("First Guess = %10.6f \r", x_guess_first);
    % fprintf("Second Guess = %10.6f \r", x_guess_second);
    % fprintf("Next guess = %10.6f \r", x_next);

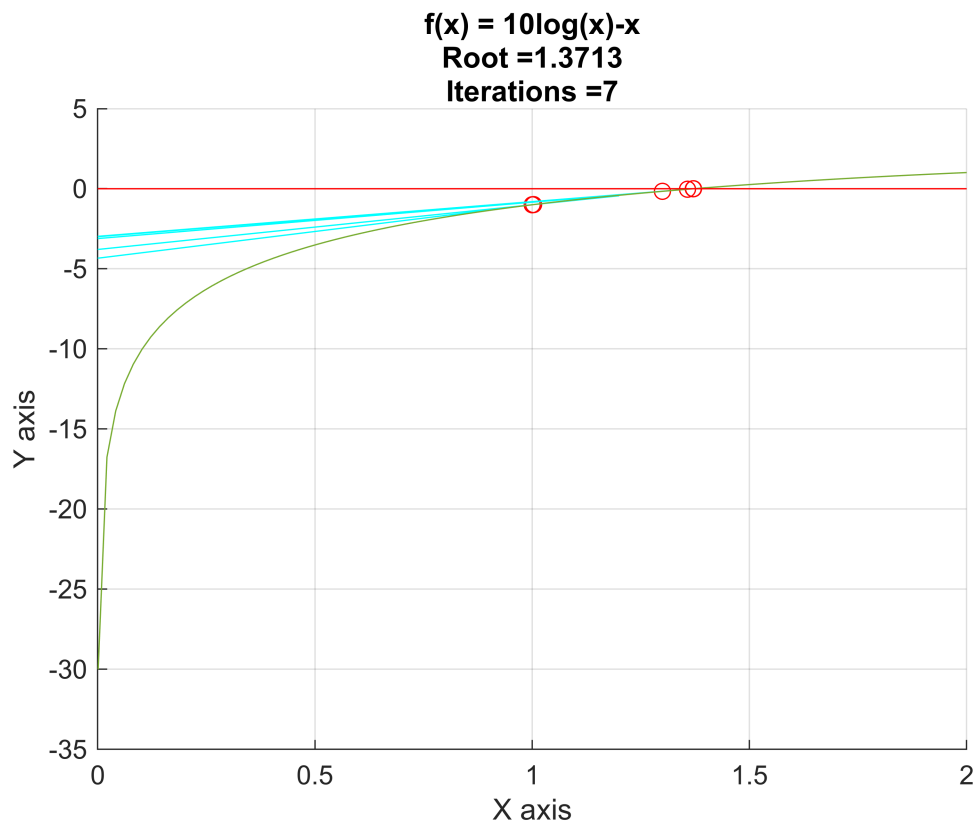
    x_limits=get(gca,"XLim");
    m = (f_at_guess_second-f_at_guess_first)/(x_guess_second-x_guess_first);
    n = -x_guess_second*m + f_at_guess_second;
    y1 = m*x_limits(1) + n;
    y2 = m*x_limits(2) + n;
    plot(x_guess_first,f_at_guess_first,'ro')
    plot(x_guess_second,f_at_guess_second,'ro')
    plot([x_limits(1) x_limits(2)], [y1 y2], 'c')

    x_guess_first=x_guess_second;
    x_guess_second = x_next;

    counter = counter+1;
end
plot(x,h_of_x)
title(num2str(x_next))

title({'f(x) = 10log(x)-x', ['Root =', num2str(x_next)], ['Iterations
=', num2str(counter)]})
grid on
xlabel("X axis")
ylabel("Y axis")
plot([0 2],[0 0], 'r') % plot y=0 axis
hold off

```



3 Newtons Intersect [6]

```
close all;
clc;
e = 2.718281828459045235360;

x = linspace(3,10,100);
tolerance = 1e-6;
max_count = 20;
counter =1;

function [value_x,value_y] = newton_(e,x)
    e = 2.718281828459045235360;
    value_x = (100 .* (e.^-x) );
    value_y = 5 .* sin( (pi.*x) ./ 2);
end

function [function_value] = newton_comb(x,e)
    e = 2.718281828459045235360;
    function_value = (100 .* (e.^-x)) - 5 * sin(pi * x / 2);
end

[value_x, value_y] =newton_(e,x); %Root [4.1949 0]
```

```

f_of_x=newton_comb(x,e);
derivative= - ((5 .* pi .* (cos((pi.*x)./2))) ./2) - (100 .* (e .^(-x)));

x_guess_first=4;
x_guess_second = x_guess_first - (f_of_x / derivative);

hold on
figure(1);

while (abs(x_guess_first-x_guess_second) > tolerance && counter < max_count)

    f_at_guess_first = newton_comb(x_guess_first);
    f_at_guess_second = newton_comb(x_guess_second);

    x_next = x_guess_second-f_at_guess_second * ( (x_guess_second -
x_guess_first) / (f_at_guess_second - f_at_guess_first));

    % fprintf("ITERATION %i \r", counter);
    % fprintf("First Guess = %10.6f \r", x_guess_first);
    % fprintf("Second Guess = %10.6f \r", x_guess_second);
    % fprintf("Next guess = %10.6f \r", x_next);

    x_limits=get(gca,"XLim");
    m = (f_at_guess_second-f_at_guess_first)/(x_guess_second-x_guess_first);
    n = -x_guess_second*m + f_at_guess_second;
    y1 = m*x_limits(1) + n;
    y2 = m*x_limits(2) + n;
    plot(x_guess_first,f_at_guess_first,'bo')
    plot(x_guess_second,f_at_guess_second,'ro')
    % plot([x_limits(1) x_limits(2)], [y1 y2], 'c')

    x_guess_first=x_guess_second;
    x_guess_second = x_next;

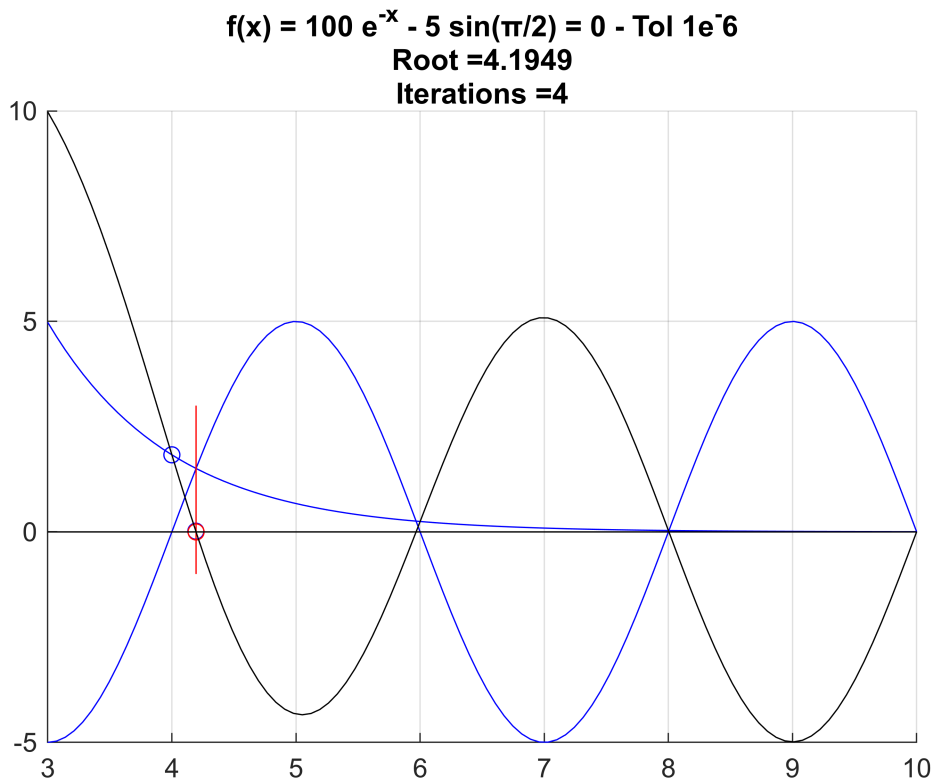
    counter = counter+1;
end

%plot(x,derivative,"g")
plot(x,value_x,"b")
plot(x,value_y,"b")
plot([3 10],[0 0],"black")
plot([x_next x_next], [-1 3],"r")
plot(x,f_of_x,"black")

title({'f(x) = 100 e^-^x - 5 sin(pi/2) = 0 - Tol 1e^-6', ['Root =',num2str(x_next)],
['Iterations =',num2str(counter)]});
grid on

hold off

```



```
close all;
clc;
e = 2.718281828459045235360;

x = linspace(3,10,100);
tolerance = 1e-10;
max_count = 20;
counter =1;

[value_x, value_y] =newton_(e,x); %Root [4.1949 0]

f_of_x=newton_comb(x,e);
derivative= - ((5 .* pi .* (cos((pi.*x)./2))) ./2) - (100 .* (e .^(-x)));

x_guess_first=4;
x_guess_second = x_guess_first - (f_of_x / derivative);

hold on
figure(1);

while (abs(x_guess_first-x_guess_second) > tolerance && counter < max_count)
```

```

f_at_guess_first = newton_comb(x_guess_first);
f_at_guess_second = newton_comb(x_guess_second);

x_next = x_guess_second - f_at_guess_second * ( (x_guess_second -
x_guess_first) / (f_at_guess_second - f_at_guess_first));

% fprintf("ITERATION %i \r", counter);
% fprintf("First Guess = %10.6f \r", x_guess_first);
% fprintf("Second Guess = %10.6f \r", x_guess_second);
% fprintf("Next guess = %10.6f \r", x_next);

x_limits=get(gca,"XLim");
m = (f_at_guess_second-f_at_guess_first)/(x_guess_second-x_guess_first);
n = -x_guess_second*m + f_at_guess_second;
y1 = m*x_limits(1) + n;
y2 = m*x_limits(2) + n;
plot(x_guess_first,f_at_guess_first,'bo')
plot(x_guess_second,f_at_guess_second,'ro')
% plot([x_limits(1) x_limits(2)], [y1 y2], 'c')

x_guess_first=x_guess_second;
x_guess_second = x_next;

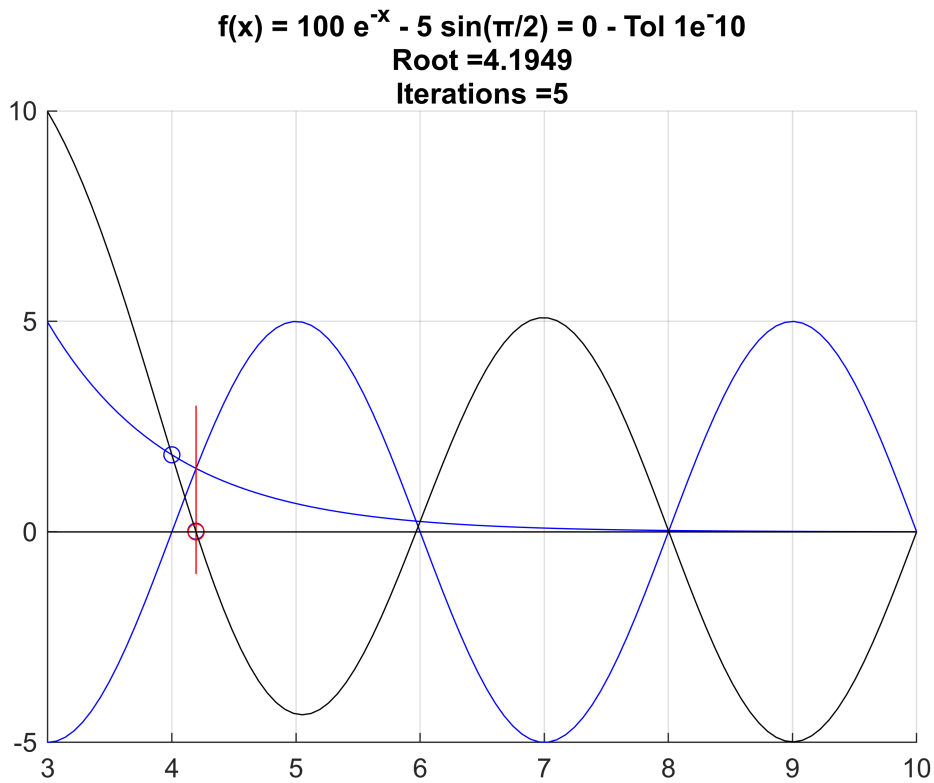
counter = counter+1;
end

%plot(x,derivative,"g")
plot(x,value_x,"b")
plot(x,value_y,"b")
plot([3 10],[0 0],"black")
plot([x_next x_next], [-1 3],"r")
plot(x,f_of_x,"black")

title({'f(x) = 100 e^-^x - 5 sin(π/2) = 0 - Tol 1e^-10', ['Root =',num2str(x_next)],
['Iterations =',num2str(counter)]});
grid on

hold off

```



4 Bracketing Bisect [6]

```
clear;
close all;
clc;

x = linspace(4,10,100);
tolerance = 1e-6;
max_count= 20;
counter = 1;

e = 2.718281828459045235360;

hold on
x_guess_first=4;
x_guess_second=x_guess_first+1;

for i= 1:10;
    f_of_x =newton_comb(x,e);
```

```

while(abs(x_guess_second-x_guess_first)>tolerance && counter<max_count) &&
x_guess_first <=11
    f_at_guess_first = newton_comb(x_guess_first);
    f_at_guess_second = newton_comb(x_guess_second);

    mid_point = (x_guess_first + x_guess_second) / 2;
    f_at_mid_point = newton_comb(mid_point);
    if x_guess_first < 9;
        plot([x_guess_first x_guess_first], [0 f_at_guess_first], "r-");
        plot([x_guess_second x_guess_second], [0 f_at_guess_second], "r-");
    end

    if f_at_mid_point * f_at_guess_second <=0;
        x_guess_first = mid_point;
    else
        x_guess_second = mid_point;
    end
    counter = counter + 1;

end

x_guess_first = x_guess_first + 1;
x_guess_second=x_guess_first+1;
if x_guess_first <=11
    % display(table(mid_point,counter))
    fprintf("Counter when diverging to zero = %10.6f \r", counter);
    fprintf("Value of the root found = %10.6f \r", mid_point);
    if 0.5>newton_comb(mid_point);
        plot(mid_point,0,"black o");
    end
end
counter = 1;
% display(table(mid_point,counter))
end

```

```

Counter when diverging to zero = 20.000000
Value of the root found = 4.194933
Counter when diverging to zero = 20.000000
Value of the root found = 5.967379
Counter when diverging to zero = 20.000000
Value of the root found = 6.967381
Counter when diverging to zero = 20.000000
Value of the root found = 8.004253
Counter when diverging to zero = 20.000000
Value of the root found = 9.999420

```

```

[value_x, value_y] =newton_(e,x); %Root [4.1949 0]

f_of_x=newton_comb(x,e);
derivative= - ((5 .* pi .* (cos((pi.*x)./2))) ./2) - (100 .* (e .^(-x)));

x_guess_second = x_guess_first - (f_of_x / derivative);

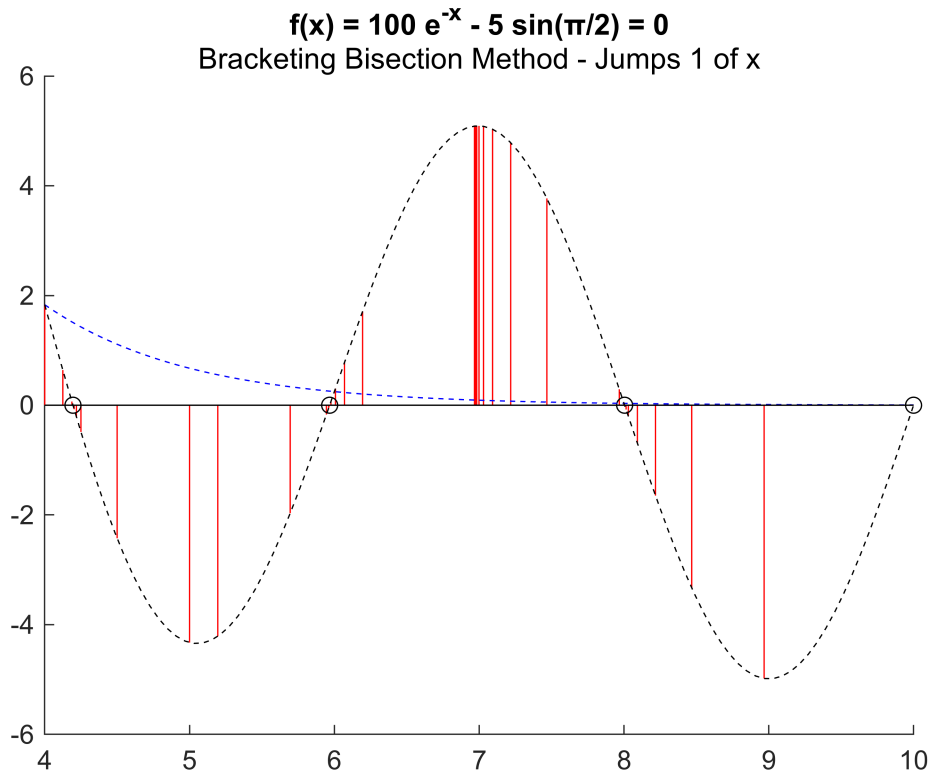
```

```

plot(x,value_x,"b","LineStyle","--")
plot(x,f_of_x,"black --");
plot([4 10], [0 0], "black");
title("f(x) = 100 e-x - 5 sin(π/2) = 0", "Bracketing Bisection Method - Jumps 1 of x")

% legend("Roots");
hold off

```



4 Bracketing Newton [6]

```

clear;
close all;
clc;
e = 2.718281828459045235360;

x = linspace(4,10,100);
tolerance = 1e-6;
max_count = 20;
counter =1;

x_guess_first=4;

```

```

for i = 1:4

    [value_x, value_y] =newton_(e,x); %Root [4.1949 0]

    f_of_x=newton_comb(x,e);
    derivative= - ((5 .* pi .* (cos((pi.*x)./2))) ./2) - (100 .* (e .^(-x)));

    x_guess_second = x_guess_first - (f_of_x / derivative);

    hold on
    figure(1);

    while (abs(x_guess_first-x_guess_second) > tolerance && counter < max_count)

        f_at_guess_first = newton_comb(x_guess_first);
        f_at_guess_second = newton_comb(x_guess_second);

        x_next = x_guess_second-f_at_guess_second * ( (x_guess_second -
x_guess_first) / (f_at_guess_second - f_at_guess_first));

        % fprintf("ITERATION %i \r", counter);
        % fprintf("First Guess = %10.6f \r", x_guess_first);
        % fprintf("Second Guess = %10.6f \r", x_guess_second);
        % fprintf("Next guess = %10.6f \r", x_next);

        x_limits=get(gca,"XLim");
        m = (f_at_guess_second-f_at_guess_first)/(x_guess_second-x_guess_first);
        n = -x_guess_second*m + f_at_guess_second;
        y1 = m*x_limits(1) + n;
        y2 = m*x_limits(2) + n;
        %plot(x_guess_first,f_at_guess_first,'bo')
        %plot(x_guess_second,f_at_guess_second,'ro')
        %plot([x_limits(1) x_limits(2)],[y1 y2],'c')

        x_guess_first=x_guess_second;
        x_guess_second = x_next;

        counter = counter+1;

    end
    x_guess_first=x_guess_first+1.5;

    fprintf("Value of iteration to divergue to zero = %1.0f \r ",counter);
    fprintf("Value of the root =%10.6f \r",x_next);

    counter = 1;

    if x_next == x_next

```

```

    %fprintf("Root = %f",x_next)'
    plot(x_next,0,"ro")
    plot([x_next x_next], [-1 3],"r")
    intersection = (100 .* (e.^-x_next));
    plot(x_next,intersection,"blue o")
    % x_next
else
end

```

```
end
```

```

Value of iteration to divergue to zero = 5
Value of the root = 4.194932
Value of iteration to divergue to zero = 6
Value of the root = 5.967379
Value of iteration to divergue to zero = 6
Value of the root = 8.004253
Value of iteration to divergue to zero = 6
Value of the root = 9.999422

```

```

plot(x,value_x,"b","LineStyle","--")
plot(x,value_y,"black","LineStyle","--")
plot([4 10],[0 0],"black")

% plot(x,f_of_x,"black")
% title(num2str(x_next));
title("f(x) = 100 e^-^x - 5 sin(π/2) = 0", "Bracketing Newtons Method - Jumps of
1.5 of x")
%legend;
hold off

```

